(Preprint) AAS 17-XXX

GLOBAL, MULTI-OBJECTIVE TRAJECTORY OPTIMIZATION WITH PARAMETRIC SPREADING

Matthew A. Vavrina,[*] Jacob A. Englander,[†] Sean M. Phillips,[‡] and Kyle M. Hughes[§]

Mission design problems are often characterized by multiple, competing trajectory optimization objectives. Recent multi-objective trajectory optimization formulations enable generation of globally-optimal, Pareto solutions via a multi-objective genetic algorithm. A byproduct of these formulations is that clustering in design space can occur in evolving the population towards the Pareto front. This clustering can be a drawback, however, if parametric evaluations of design variables are desired. This effort addresses clustering by incorporating operators that encourage a uniform spread over specified design variables while maintaining Pareto front representation. The algorithm is demonstrated on a Neptune orbiter mission, and enhanced multidimensional visualization strategies are presented.

INTRODUCTION

Almost inherently, mission design is a compromise between different objectives such as maximizing payload mass and minimizing the time of flight (TOF), among others. Optimizing multiple objectives to produce the Pareto front[1] of optimal solutions, and thus, an understanding of the tradeoff between the objectives, is valuable in preliminary assessment of mission feasibility and subsequent design refinement. Moreover, the ability to evaluate the sensitivity of the objectives to the mission design variables such as launch date, arrival date, destination body (e.g., asteroid-hopping missions), and mission systems variables such as solar array size, the number of low-thrust engines, and launch vehicle is frequently sought. When conducting these trade studies, identifying the global optimum can be the difference between mission viability and one that does not close within requirements.

Recent work on global, multi-objective trajectory and systems optimization using evolutionary algorithms (EA) has enabled simultaneous optimization of multiple objectives to efficiently generate Pareto front representations.[2,3,4,5] One such approach is implemented in NASA's Evolutionary Mission Trajectory Generator (EMTG),[6] which incorporates a global-local hybrid algorithm to solve multi-objective hybrid optimal control problems (MOHOCP).[4,5] EMTG is composed of an inner and outer loop that function in concert for global, multi-objective optimization of either high- or low-thrust trajectories. Both high- and low-thrust trajectory transcriptions are formulated as non-linear programming problems, and are available for problem definition within the EMTG inner-loop formulation. A NLP solver, SNOPT,[7] is applied to locally optimize an initial guess generated by a monotonic basin hopping (MBH) routine, providing a global, stochastic search of the trajectory design space. An outer-loop, multi-objective optimizer based on the non-dominated sorting genetic algorithm II (NSGA-II)[8] drives the inner loop to identify a set of Pareto-optimal solutions. The outer loop

[*] Senior Systems Engineer, a.i. solutions, Inc., 4500 Forbes Blvd., Suite 300, Lanham, MD, 20706, USA.
[†] Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center, Greenbelt, MD, 20771, USA.
[‡] Principal Software Engineer, a.i. solutions, Inc., 4500 Forbes Blvd., Suite 300, Lanham, MD, 20706, USA.
[§] Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center, Greenbelt, MD, 20771, USA.

conducts a search over user-defined trajectory variables such as TOF, launch date, arrival date, and flyby/gravity-assist bodies. Additionally, the outer loop is capable of mission systems optimization, with spacecraft hardware parameters such as engine type, number of thrusters, solar array power, solar array type, and launch vehicle included as discrete design variables. The NLP/MBH inner loop optimizes the continuous NLP trajectory variables for the set of system and trajectory parameters defined by the outer loop. The hybrid algorithm strategically steers the EA population of designs towards the Pareto front for any number objectives such as maximizing final spacecraft mass or minimizing TOF, departure C_3 power, or total ΔV.

A natural byproduct of EA formulations, such as the outer loop in EMTG, is that clustering in design space can occur in evolving the population of designs towards the Pareto front and the optimal objective space. That is, members of the population that occupy optimal regions of the design space will tend to have better objective function values, and will be promoted to future EA generations. For some problem types and objective functions, the optimal regions of the design space are distinguished by narrow spans of design variable values, which induces the clustering in design space. This clustering can be a drawback, however, if parametric evaluations of principal design variables such as launch and arrival date are desired. That is, a uniform spreading across the range of the design parameter is frequently sought at some point in the mission design process. Understanding the sensitivity of the mission objectives to critical design variables can provide important mission insight such as how long a launch can be delayed until the mission become infeasible.

This work addresses the intrinsic design space clustering of EA-based multi-objective optimization algorithms by developing EA operators that encourage a uniform spread over specified design variables while concurrently maintaining a representation of the Pareto front. The parametric spreading algorithm is based on the previously-developed NSGA-II formulation for global, multi-objective trajectory and systems optimization in EMTG. The first section of the paper provides background on the high- and low-thrust trajectory transcriptions, before outlining the MBH/NLP-based inner-loop algorithm applied for single objective trajectory optimization. Next, the parametric spreading, multi-objective algorithm is detailed with description of the operators developed to encourage spreading in design space. Finally, results of the algorithm are applied to two example problems, a high-thrust Neptune rendezvous mission and a low-thrust mission to return a sample from Deimos.

TRAJECTORY MODELING AND OPTIMIZAITON

EMTG exploits a multi-layer mission architecture that allows for flexible optimization of multiple flyby trajectories with variable trajectory legs. The multiple-shooting framework incorporated in the high- and low-thrust transcriptions affords the inner loop robust optimization properties. This robustness is critical to the monotonic basin hopping routine that stochastically searches for optimal trajectories, providing a NLP solver initial guesses for gradient-based refinement. In turn, the flexibility, efficiency, and robustness of the inner loop is vital to outer-loop multi-objective optimization efficacy.

Mission Structuring

EMTG missions are formulated with three levels of event types as illustrated in Figure 1. The top-level mission structure includes all event types comprising a mission such as departures, arrivals, Deep Space Maneuvers (DSMs), and flybys. Missions consist of one or more journeys, defining the trajectory in terms of the set of events at the required target bodies of the mission: the starting, ending, and any required intermediate bodies. Events are defined by the user or the outer loop. The boundaries of a journey are the locations at which the spacecraft will execute specific events, and can be constrained in numerous ways. Journeys are in turn made up of one or more phases. Phases are similar to journeys, except that they may start and end at bodies other than the required journey bodies to enable flybys

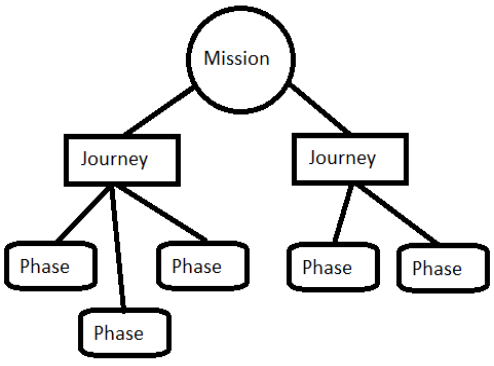

Figure 1. EMTG mission structure

that are incorporated strictly for trajectory performance, and not as encounters that are required as part of the mission. As an example, the DAWN mission to Vesta and Ceres would be composed of two journeys: an initial Earth to Vesta journey and a Vesta to Ceres journey. However, the initial journey incorporates a Mars flyby to decrease total ΔV, and two phases comprise that initial journey. This three-level structure allows an optimizer to vary the number of phases as well as the flyby bodies within a journey, and is applied for all transcriptions in EMTG.

High-Thrust Trajectory Modeling

For high-thrust trajectory optimization, this work employs a direct, multiple shooting transcription within a patched-conic modeling framework called MGA*n*DSM*s* for multiple gravity assists (MGA) with *n* DSMs per phase, using a shooting technique, *s*.[9] The MGA*n*DSM*s* transcription enables the design of a wide variety of mission scenarios with any number of DSMs, multiple gravity assists, and a range of mission-specific constraints directly prescribed in the optimization formulation. Within the inner loop the MGA*n*DSM*s* exploits analytic derivatives for improved robustness to initial guesses.

A diagram of the trajectory transcription is illustrated in Figure 2, outlining the trajectory structure for a single phase in a mission. A match point is incorporated between the two end points in each trajectory phase with forward/backward shooting for control of initial guess error growth. The endpoints represent the beginning and end of the phase in mission time (left to right in figure), and are typically planetary bodies. The right end point can be a target body, flyby body, or gravity assist body. The arrows in the diagram represent impulsive maneuvers, which instantaneously change the spacecraft's velocity. The maneuvers are separated in time by a Δt optimization variable, and maneuvers at the phase end points are also possible if designated. The number of maneuvers per phase may be specified by the user or the outer loop. If fewer than *n* maneuvers are optimal for the transfer, one or more of the potential maneuvers is driven to a zero magnitude.

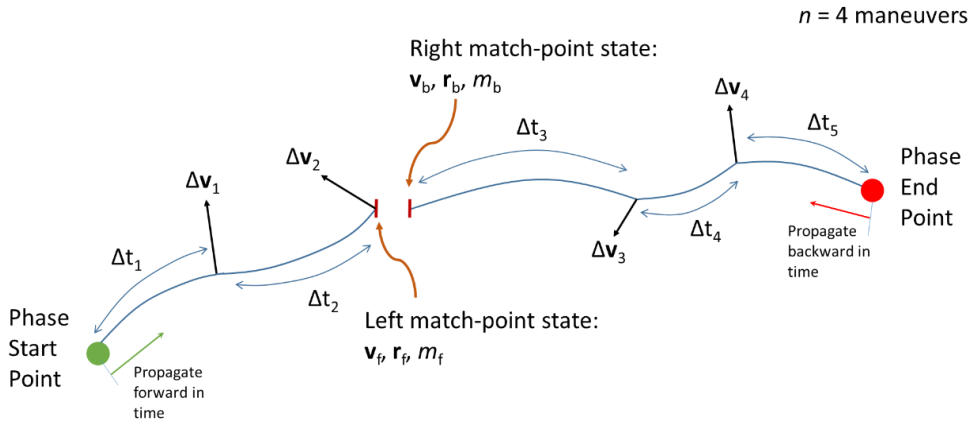

Figure 2. MGA*n*DSM*s* transcription diagram with *n*=4

From the phase start point, the trajectory is propagated forward in time until the match-point DSM is reached. At each maneuver epoch, the spacecraft velocity is changed instantaneously by the impulsive maneuver vector as determined by the optimizer. Similarly, the trajectory is propagated backwards in time from the phase end point to the match point with each maneuver instantaneously modifying the spacecraft velocity along the way. Kepler propagation is used by the propagator for efficiency. A match point maneuver resides on the forward propagation side of the match point, and varies in time according to the Δt variables of the phase. For a feasible trajectory the optimizer must drive any discontinuity in position, velocity, or mass within a small tolerance at the match point. Additionally, the summation of the Δt variables of the phase must be equal to the phase time of flight. If a phase begins with a launch vehicle, a polynomial curve fit is applied to determine initial spacecraft mass, m_0, as a function of C_3:

$$m_0 = (1-\sigma_{LV})(a_{LV}C_3^5 + b_{LV}C_3^4 + c_{LV}C_3^3 + d_{LV}C_3^2 + e_{LV}C_3 + f_{LV} \quad (1)$$

where σ_{LV} is a user-specified launch vehicle margin, and a_{LV} through f_{LV} are the polynomial coefficients derived from a launch vehicle performance curve.

The full mission transcription applies a multiple shooting strategy from body-based control nodes to the phase match points as illustrated in Figure 3. In this way, errors in the initial guess do not grow unabated throughout a full mission propagation. Multiple trajectory phases between flybys are constructed for multiple gravity assist problems with n maneuvers allowed for each phase. For simplified modeling, flybys are modeled using the zero sphere-of-influence approach[10] in which the spacecraft position is matched to the flyby body at a control node, and the body imparts an instantaneous change in the spacecraft's central-body velocity.

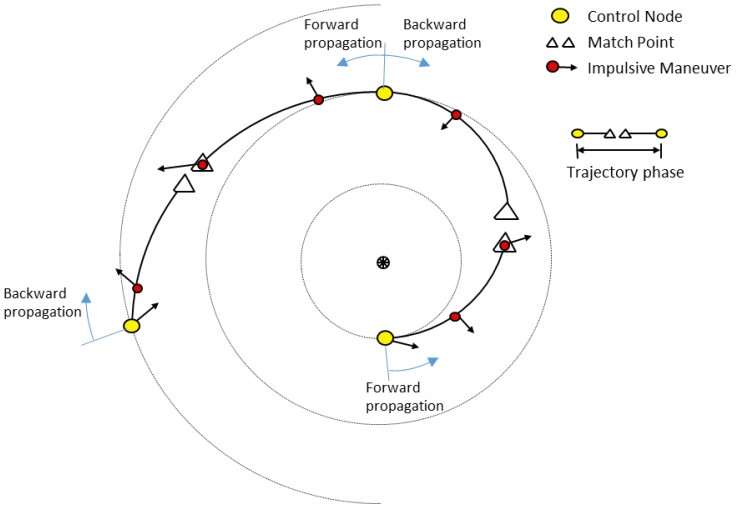

Figure 3: MGA*n*DSM*s* full-mission transcription

Low-Thrust Trajectory Modeling

As with the high-thrust trajectory modeling, the low-thrust trajectory transcription applied in this work is structured on a robust multiple shooting scheme. Namely, the Sims-Flanagan[11] transcription is employed, in which low-thrust arcs are discretized into a user-specified number of segments. At the midpoint of each segment an impulsive ΔV maneuver is applied, approximating continuous thrusting by the spacecraft. The ΔV magnitude is constrained by the maximum change in velocity that could be accrued at maximum thrust during the segment. This direct method of transcription significantly reduces the dimensionality of the problem for optimization efficiency at the cost of some modeling fidelity. Further improving efficiency is the use of two-body propagation between segment midpoints, which substantially reduces computational complexity associated with higher-fidelity propagation options. The spacecraft mass is propagated assuming a constant mass flow rate throughout the time segment.

Like the high-thrust transcription, a forward/backward shooting technique is applied to a match point located between two end point bodies as illustrated in Figure 4. For feasibility, the position and velocity discontinuities at the match point must be reduced to a small tolerance by the optimizer. This approach to low-thrust trajectory optimization has been applied in wide ranging mission studies using several different software packages including MALTO[12], GALLOP[13], and Parallel Global Multiobjective Optimizer,[14] in addition to EMTG.

Accurate hardware modeling is important for low-thrust trajectories as the trajectory is tightly coupled with the spacecraft mass and thrust capability. Furthermore, the outer loop in this work searches over discrete hardware models driving both the mass of the vehicle and the available thrust. Polynomial fits are incorporated to model

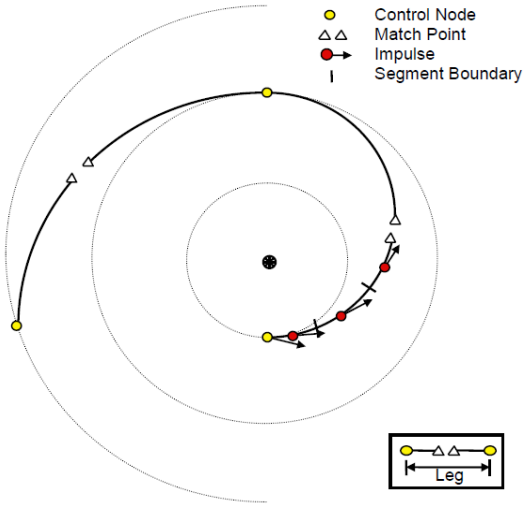

Figure 4. Sims-Flanagan trajectory formulation (from Reference 12)

the performance of the thruster. The thrust, T, in mN, and mass flow rate, $\dot{m}_{\max}$, in mg/s for each electric propulsion thruster are modeled as a function of the current power to the engine such that

$$T = a_T P^4 + b_T P^3 + c_T P^2 + d_T P + e_T \tag{2}$$

and

$$\dot{m}_{\max} = a_F P^4 + b_F P^3 + c_F P^2 + d_F P + e_F \,. \tag{3}$$

The power available to an engine, P, is limited between an upper and lower bound dependent on the on the thruster. The thrust is zero if the supplied power to the thruster is less than the minimum operational power level, and power available to the engine is capped at the maximum power level. The polynomials are, in general, based on curve fits to laboratory data. Different curve fits may be available for the same engine based on selected operating modes to which the curves are fit. EMTG allows for any number thrusters. For the example problems in this paper a $1/r^2$ power model is applied, where r is the range to the Sun. However, EMTG supports higher fidelity polynomials according to specific array performance. The launch vehicle is modeled as in equation 1 for the high-thrust transcription.

Inner Loop: Global Single-Objective Optimization

The NLP problem resulting from both MGA*n*DSM*s* and the Sims-Flanagan transcription for a single objective can be stated as:

$$\begin{aligned} \text{minimize:} \quad & f(\mathbf{x}) \\ \text{subject to:} \quad & \mathbf{c}(\mathbf{x}) \le 0 \\ & A\mathbf{x} \le 0 \\ & \mathbf{x}_{lb} \le \mathbf{x} \le \mathbf{x}_{ub} \end{aligned} \tag{4}$$

where $f(\mathbf{x})$is the objective function, $\mathbf{c}(\mathbf{x})$ is a vector of the nonlinear inequality constraints, A is a matrix of linear constraints, and $\mathbf{x}_{lb}$ and $\mathbf{x}_{ub}$ are vectors defining the lower and upper bounds on the vector of problem decision variables $\mathbf{x}$.

The availability of analytic derivatives and a sparse Jacobian make the sequential quadratic programming (SQP) solver SNOPT well suited to optimize the inner NLP problem. Exploiting derivative information for a local search of the design enables efficiency and robustness, but an initial guess in the vicinity of the local optimum is still required by the SQP routine. Additionally, SQP approaches do not provide a global search of the design space. To overcome these characteristics of the gradient-based approaches, stochastic optimization wrappers can be implemented to globally search the design space without requiring an initial guess in a global-local hybrid formulation.[15] In these approaches the global search algorithm guides the local search layer, tactically providing the inner loop an initial guess for gradient-based local refinement.

In this work a monotonic basin hopping strategy is implemented as a global search component in EMTG. MBH operates as a

Algorithm 1 Monotonic Basin Hopping with NLP

generate random point $\mathbf{x}$
run NLP solver to find point $\mathbf{x}^*$ using initial guess
$\mathbf{x}_{current} = \mathbf{x}^*$
if $\mathbf{x}^*$ is feasible **then**
  save $\mathbf{x}^*$ to archive
while not hit stop criterion **do**
  generate $\mathbf{x}'$ randomly perturbing $\mathbf{x}_{current}$
  for each TOF variable t_i in x' **do**
    if rand(0,1) < $\rho_{time\text{-}hop}$ **then**
      shift t_i forward or backward 1 synodic period
  run NLP solver to find point $\mathbf{x}^*$ from $\mathbf{x}'$
  if $\mathbf{x}^*$ is a feasible **and** $f(\mathbf{x}^*) < f(\mathbf{x}_{current})$ **then**
    $\mathbf{x}_{current} = \mathbf{x}^*$
    save $\mathbf{x}^*$ to archive
  else if $\mathbf{x}^*$ is infeasible and $\|c(\mathbf{x}^*)\| < \|c(\mathbf{x}_{current})\|$ **then**
    $\mathbf{x}_{current} = \mathbf{x}^*$
return best $\mathbf{x}^*$ in archive

Monte-Carlo-like optimization scheme, taking stochastic "hops" in design space from an initial base design. The approach is well-suited for problems in which local optima are clustered together as the search is focused on the region of the design space near the current best solution (the exploitation element of the algorithm). Less frequent larger hops from the base design, provide broader exploratory capacity. MBH does not require an initial guess, and is generally started from a random initial base design. The MBH+NLP algorithm is summarized in Algorithm 1.[6]

Multiple objective functions are available for selection in the inner loop. Maximizing the final mass or minimizing the TOF are most typically applied. Other possible objective functions include minimization of total mission ΔV, minimization or maximization of the launch or arrival date within bounded windows, and minimization of flight time, among other options.

MULTI-OBJECTIVE OPTIMIZATION WITH PARAMETRIC SPREADING

The inner loop provides an efficient means to global optimization for single objective function for both high- and low- thrust missions. The outer loop, in turn, enables multi-objective optimization, allowing for variation in parameters such as TOF, launch date, arrival date, number of electric propulsion thrusters, and thruster type for any number of objectives. The outer loop applies an EA-based multi-objective algorithm to define and steer the inner-loop, which is structured as a nested loop. No user-defined initial guess is needed to start the optimization. In this work, the EA-based outer loop is developed to not only provide a set of equally-optimal multi-objective solutions, but also to generate a representation of the sensitivity of the objectives to specified outer loop design variables.

Multi-Objective Optimization Problem Statement

With multiple optimization objectives, the aim of the optimization process is the generation of the Pareto front of solutions. The Pareto front represents the set of equally-optimal, compromise solutions in which no improvement can be achieved in one objective without degrading at least one other objective. In objective function space, the Pareto front forms a hyper-surface of equal dimension to the number of objectives. The Pareto front represents all possible Pareto-optimal solutions, and, in general, is not known *a priori*. A multi-objective optimization routine must then generate numerous Pareto optimal solutions to represent the Pareto front and enable tradeoff decisions between the objectives. The Pareto front for a notional low-thrust trajectory optimization problem with the objectives to maximize final spacecraft mass and minimize time of flight is illustrated in Figure 5.

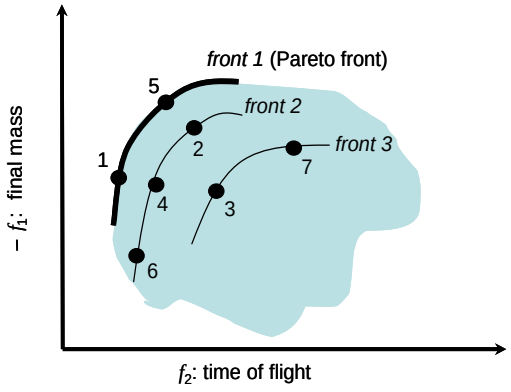

Figure 5. Two-dimensional, non-dominated fronts of a multi-objective trajectory optimization example.

The multi-objective optimization problem can be stated as follows:

$$\begin{aligned} &\text{minimize:} \quad \mathbf{f}(\mathbf{x}) \\ &\text{subject to:} \quad x_i^L \le x_i \le x_i^U \qquad i = 1,\ldots,n_{obj} \end{aligned} \tag{1}$$

where **f(x)** is a vector of objectives

$$\mathbf{f}(\mathbf{x}) = [\, f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad \cdots \quad f_{n_{obj}}(\mathbf{x})]^{\mathrm{T}}, \tag{2}$$

$\mathbf{x}$ is a vector of design variables (with $\mathbf{x}^L$ and $\mathbf{x}^U$ lower and upper bounds), n_{obj} is the scalar number of objectives. Note that no constraints are incorporated in this multi-objective formulation. The objective space is n_{obj}-dimensional, and the objective functions are often coupled (containing the same design variables) and competing (the optimal solution in one objective is not the same optimal solution in the other objectives).

The multi-objective optimization concept of *domination* allows for the comparison of a set of designs with multiple objectives, providing a measure of the relative quality of the design. When comparing two multi-objective designs, the design $\mathbf{x}_1$ *dominates* design $\mathbf{x}_2$ if:

$$\begin{aligned} &\forall p : f_p(\mathbf{x}_1) \le f_p(\mathbf{x}_2) \qquad p = 1,2,\ldots,n_{obj} \\ &\qquad\qquad \textit{and} \\ &\exists p : f_p(\mathbf{x}_1) < f_p(\mathbf{x}_2) \qquad p = 1,2,\ldots,n_{obj} \end{aligned} \tag{3}$$

That is, $\mathbf{x}_1$ dominates design $\mathbf{x}_2$ if, for all objectives, $\mathbf{x}_1$ is better than or equal to $\mathbf{x}_2$, *and* $\mathbf{x}_1$ outperforms $\mathbf{x}_2$ for at least one objective. In a direct comparison of two designs, if one design dominates another, that design is closer in proximity to the Pareto front. If neither design dominates the other, the designs are non-dominant to each other. Therefore, in a set of designs, the superior designs are those that are not dominated by any other design in the set, and are termed the non-dominated subset. It follows that any Pareto-optimal design is a member of the non-dominated subset associated with the entire feasible objective space and is located along the Pareto front.

Traditional approaches to multi-objective optimization such as the weighted sum[16] and ε-constraint[17] methods have several drawbacks. Many techniques necessitate solving many different optimization problems via independent runs. Classical multi-objective optimization approaches can also require tuning of specialized parameters to identify solutions that are Pareto-optimal. Furthermore, some techniques are not capable of producing Pareto fronts that are non-convex, and thus important optimal solutions may not be identified. To address some of the difficulties of multi-objective optimization, specialized genetic algorithms[18] (GA) have been developed to generate representations of the Pareto front of solutions.[19]

Design Space Clustering in Multi-Objective Evolutionary Algorithms

A GA is a stochastic search and optimization technique that simulates biological natural selection and reproduction with the goal of identifying globally-optimal designs. This class of algorithm functions with a population of designs, and probabilistic transition rules are generally executed by three genetic operators: selection, crossover, and mutation. The operators are applied to a parent population to produce a new offspring generation that is better adapted to fitness landscape defined by the optimization formulation. A key advantage of GAs is that initial guesses are not required as the initial population is randomly defined. Additionally, without the need for gradient information, the genetic operators can effectively explore discrete, multimodal, and expansive design spaces. Furthermore, modifications to the simple GA can transform the algorithm into an effective multi-objective optimization scheme.

GAs have the benefit of operating with an entire population of designs, and this population can be evolved towards the Pareto front in a single optimization run. The ability to operate on many designs simultaneously can enable improvements in efficiency and the capacity to generate a broad and uniform representation of the Pareto front versus methods that require many repetitions of single-objective optimization routines. An effective multi-objective approach developed by Deb called the non-dominated sorting genetic algorithm (NSGA) uses a genetic algorithm in which the fitness of an individual in the population is based on its relative proximity to the population's non-dominated front.[20] With a single measure defining the fitness of an individual in the population, the genetic operators of the GA function to evolve the population toward the globally-optimal Pareto front in the same way that the population of a single-objective GA evolves toward the globally optimal solution. A second-generation non-dominated sorting genetic algorithm, the NSGA-II, improves upon the original NSGA.[20] The NSGA-II incorporates mechanisms that ensure that elite individuals from the population are retained as the population evolves. Additionally, the NSGA-II employs strategies that aim to produce a uniform representation of designs along the Pareto front.

Evolutionary preference in the base NSGA-II algorithm is provided to members of the population that are optimal in objective space. As such, the algorithm is designed to effectively ignore regions of the design space that do not map to the optimal regions of objective space as the evolution proceeds. In later generations of the GA, most individuals in the population have congregated in the optimal design space with the mutation operator providing exploration capacity outside of the design space clustering at a low rate. While this strategy is effective for generating the Pareto front, no mechanisms function to provide sensitivity of the objectives to design variable variation in a parametric sense as the population is clustered. The ability to spread

individuals in the population uniformly in design space such that both the Pareto front and a parameter trade are generated in a single run would be beneficial to the mission designer. Such a capability would save time spent on developing subsequent sensitivity studies for key design variables.

The clustering characteristic of EAs is illustrated in Figures Figure 6 and Figure 7 for a low-thrust asteroid boulder return mission with two objectives: maximize the boulder return mass and minimize the time of flight. Figure 6 illustrates the representation of the Pareto front allowing for a mission designer to trade the two objectives and providing key insight such as the low cost associated with increasing the time of flight slightly to achieve significant gains in boulder return mass around flight times of 1700 days. The red points are non-dominated and the blue are dominated. Figure 7 shows the boulder return mass objective versus the Earth arrival date design variable. The red diamonds correspond to the solutions in the final generation of the EA without employment of parametric spreading, and the blue line indicates the actual, parametric curve of return mass versus arrival date that can be used to fully understand the return mass versus arrival date sensitivity. The arrival date design variables cluster around optimal date ranges instead of spreading out across the arrival date variable bounds without a spreading mechanism in the genetic algorithm.

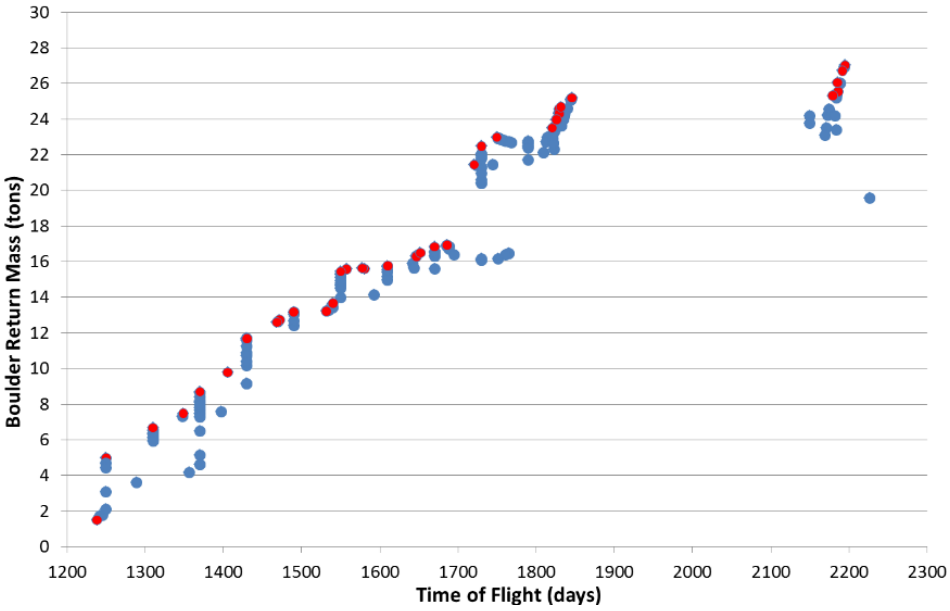

Figure 6. Two-dimensional representation of Pareto front for an asteroidal boulder return mission (maximize boulder mass and minimize time of flight)

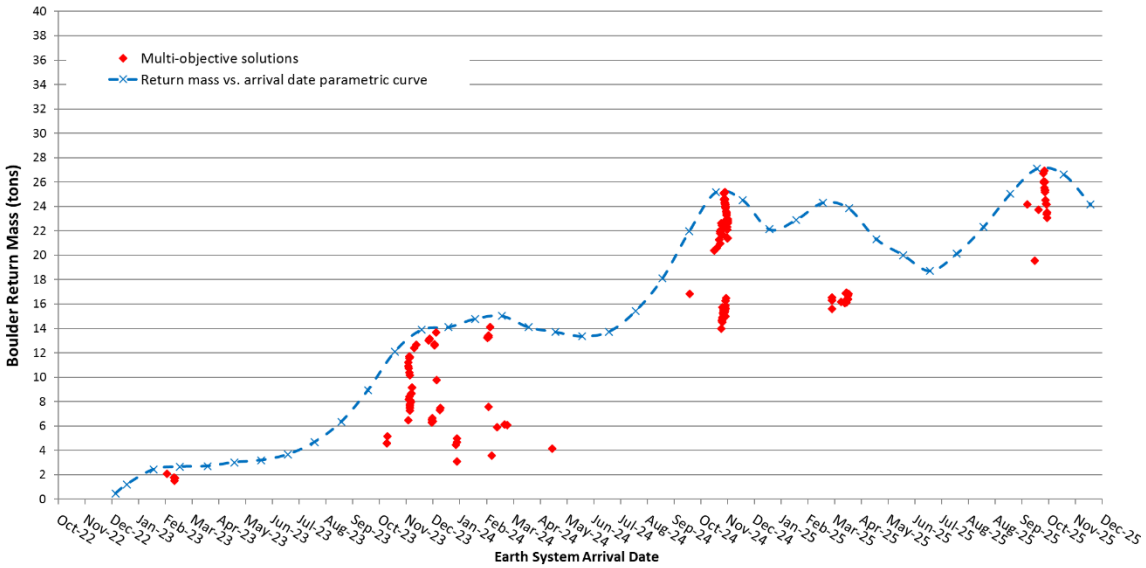

Figure 7. Boulder return mass objective function values versus Earth arrival date design variable.

Parametric Spreading Multi-Objective Genetic Algorithm

To enable parametric spreading in the outer loop of EMTG, the NSGA-II algorithm is modified to encourage diversity in design space for each specified outer-loop parameter. As with the base NSGA-II, a primary goal of the multi-objective optimization algorithm is still development of a representation of the Pareto front, however design-space diversity preservation mechanisms are incorporated in variations of the standard NSGA-II operators. Specifically, the non-dominated sorting routine that ranks individuals of the population based on their relative proximity to the Pareto front is modified such that the ranking of an individual also accounts for its relative proximity to other members in design space. Additionally, the objective space crowding distance measure of the base NSGA-II is altered so that objective space and design space crowding is accounted for in the genetic operators that evolve the population.

The base NSGA-II algorithm follows a process similar to a single-objective genetic algorithm with some key variations. Initially, a random population of designs is generated, and objective function values are assigned. The individuals are then processed through a selection operator that serves as a survival-of-the-fittest function, to determine which individuals should be allowed to pass on to future generations of the GA. In a simple GA, the metric to determine the fittest individual is simply the single objective function value. In the NSGA-II, however, there are multiple objectives to account for. The NSGA-II, instead, conducts a non-dominated sorting of the individuals in the population to rank each individual according to its relative "distance" to the Pareto front.

With a fitness assigned, members of the population most fit for the objective-space landscape are retained to form a new parent population, while those with the worst fitness are discarded Darwinianly. In tournament selection, two individuals are selected randomly from the population and their fitness is compared. The individual with the better fitness is maintained in the parent population. If two individuals in the NSGA-II have the same fitness values, the crowding distance measure breaks the tie such that the more remote individual in objective space is passed to the parent population. Crowding distance in the base NSGA-II is assigned to each individual based on the perimeter of the hyperrectangle formed by the values of two adjacent designs in objective space. This crowding-distance-based tournament selection is the first objective-space diversity preservation mechanism utilized by the NSGA-II. After tournament selection, a parent population the size of the initial population has been formed.

The parent population is then passed to a crossover operator, which randomly combines parents in the population to generate a new offspring population with a mixture of design-space (genotypic) characteristics from the parent individuals. The offspring population is randomly mutated with small variations in the design space encoding of some of the offspring individuals. Objective function values are then assigned to the new offspring population. In the EMTG outer-loop, objective function assignment is through optimization in the inner loop.

To ensure the best individuals in the population are not lost from generation to generation, elitism can be employed. In a simple GA, the top performing individuals from the last generation replace the current generation's worst fit individuals. Elitism is not as straightforward in a multi-objective GA given that there is more than one objective. To incorporate elitism in the NSGA-II, the new offspring population is combined with its originating parent population for a combined population that is twice the nominal population size. That is, if the original population is size N, the combined population is size $2N$. Again, non-dominated sorting is conducted to rank individuals in the combined population based on their non-dominated rank. Subsequently, the objective-space crowding distance is assigned. The best performing individuals in the combined population are then filtered into a new parent population of size N. The individuals in the first non-dominated front are afforded first inclusion in the new parent population, and filling of the N available slots continues according the fitness rank until the remaining number slots available is less than the number of individuals in the next batch of individuals with the same fitness rank. To fill the remaining available slots, the crowding distance of the individuals is again used to break any tie of equivalent fitness. At this point, a new parent population is formed, and the GA and NSGA-II procedure repeats.

One primary change to the base NSGA-II for parametric spreading is a modification of the non-dominated sorting routine. To promote individuals associated with the least-clustered regions of the design variables designated for parametric spreading, a parametric spreading sparsity metric is included in the non-dominated assessment of the population. In effect, the remoteness in design space for designated parametric spreading variables is treated as an additional optimization objective, and afforded the same evolutionary weight as any of the strict mission objectives. Prior to performing a non-dominated sort, the entire population is assigned a parametric sparsity in a similar fashion to the nominal objective-space crowding distance assignment, as outlined in Algorithm 2. A hyper-rectangle is formed in the parametric spreading design space about each individual such that the two neighboring individuals in design space form the vertices. The parametric sparsity is then the sum of the sides of the rectangle, with each side normalized by the range in values of the corresponding spread variable across the population. If a single variable is designated for parametric spreading, the parametric sparsity is the normalized difference in the variable values of the two neighboring individuals.

Algorithm 2 Parametric sparsity assignment

for each spread variable v
  I_v = *sort*(I, v) (sort set I in ascending order to create set I_v)
  $L = \emptyset$ (initialize solutions with variable v equal to min. value)
  $H = \emptyset$ (initialize solutions with variable v equal to max. value)
  for each solution i in I_v
    if $v = 0$ **then**
      $I_v[i]_{distance} = 0$
    if $X_v[i] = X_v^{\min}$ **then**
      $L = L \cup I_v[i]$
    else if $I_v[i]_{objectiveValue} = X_v^{\max}$ **then**
      $H = H \cup I_v[i]$
    else
      $I_v[i]_{distance} = I_v[i]_{distance} +$
        $(X_v[i+1] - X_v[i-1])/(X_v^{\max} - X_v^{\max})$
  for each variable w except v
    L_w = *sort*(L, v)
    $L_w[0]_{distance} = \infty$
    $L_w[end]_{distance} = \infty$
    H_w = *sort*(H, w)
    $H_w[0]_{distance} = \infty$
    $H_w[end]_{distance} = \infty$
  if spreading variable elitism = *true*
    for each objective p
      for each solution i in I_v
        if $f_p(I_v[i]) = f_p^{\min}$
          $I_v[i]_{distance} = \infty$
          break

As an option, the user can specify that the algorithm apply elitism for each discrete design variable encoding. If parametric elitism is selected, the individual with the best objective function value for each objective and each discrete design variable value is assigned a high parametric sparsity to ensure it is not nominated during the non-dominated sort. Care must be taken, however, when applying this option as the number of individuals provided elite status can be large and overwhelm the population if there are numerous objectives or a large number of possible parametric spreading discrete values. Nonetheless, for appropriate problems, the elitism option can provide stronger evolutionary pressure to parameter spreading at the cost of evolving the population less efficiently towards the Pareto front.

Algorithm 3 Non-dominated sort for parametric spreading

for each individual m in population P
  $S_m = \emptyset$ (initialize set of solutions dominated by m)
  $c_m = 0$ (initialize number times m is dominated to 0)
  for each individual n in population P
    if (m dominates n for all objectives & parametric sparsity) **then**
      $S_m = S_m \bigcup \{n\}$ (add n to S_m)
    else if (n dominates m for all objectives & parametric sparsity) **then**
      $c_m = c_m + 1$ (increment when m is dominated)
  if $c_m = 0$ **then**
    $m_{rank} = 1$ (m is member of 1st non-dominated front)
    $F_1 = F_1 \bigcup \{m\}$
$i = 1$
while $F_i \neq \emptyset$
  $Q = \emptyset$ (initialize temporary set Q to the empty set)
  for each individual m in front F_i
    for each individual n in S_m
      $c_m = c_m - 1$
      if $c_m = 0$ **then**
        $n_{rank} = i + 1$
        $Q = Q \bigcup \{n\}$ (individual n belongs to i+1 front)
  $i = i + 1$
  $F_i = Q$ (front i is set equivalent to set Q)

The non-dominated sorting algorithm for parametric spreading incorporates an additional general diversity preserving mechanism from the base NSGA-II to provide further exploration properties. When evaluating domination, any duplicates of an individual in the population are automatically cataloged as dominated so that they do not reside in the first non-dominated front and receive the top fitness value. This mechanism is beneficial for parametric spreading in the modified NSGA-II as top performing individuals can come to overwhelm the best non-dominated front in later generations. Promoting duplicates of top-performing individuals is a strategic characteristic of the base NSGA-II, and is generally beneficial for algorithm efficiency when solely aiming for multi-objective optimization. However, when additionally striving for a uniform parametric spread over the specified design variables, diversity in the population is of critical importance. The non-dominated sorting for parametric spreading is detailed in Algorithm 3.

Another fundamental departure from the base NSGA-II for parametric spreading is the crowding distance metric that is applied for breaking ties in the tournament selection operator and filling the final slots of the N-sized parent population. As both objective space crowding and parametric crowding in design space is important in this implementation, a combined crowding distance metric is utilized to account for both considerations. The combined crowding distance provides diversity preservation by promoting the individuals in the population that reside in the least dense regions of objective and parametric space in an averaged sense.

For determination of the combined crowding distance, first a modified objective-space crowding distance as defined in Reference 4 is assigned to each individual in the set of designs undergoing assignment. Next, a parametric sparsity value is assigned to each individual in the set according to Algorithm 2. Individuals at the extreme values of the parametric spreading values in the set are assigned a large parametric sparsity. If there is a group of individuals at the extreme values, only select members of the boundary group are preferred. Specifically, for each parametric spreading variable the individuals in the boundary group with the minimum and maximum values for the other parametric spreading variables are assigned large values. Given that there can be a different number of spreading variables and objectives, the spreading variable component of the combined crowding distance is weighted according to ratio of designated spreading variables values to objectives. This weighting gives equal preference to design and objective space spreading as the population evolves. Finally, with both an objective-space crowding distance and parametric sparsity, the two values are combined into the single combined crowding distance metric.

The full outer-loop algorithm is illustrated as a flow chart in Figure 8. The parametric spreading multi-objective GA generates problems for the refinement in the inner-loop MBH/NLP algorithm based on the objectives defined by the user and a menu of discrete design variables. Trajectory variables in the outer loop such as launch date, arrival date and flight time are incorporated via a cap-and-optimize process, bounding the inner-loop trajectory problem. Other outer-loop variables provide strict definitions for the mission such as the number of thrusters and the flyby sequence. A variable flyby sequence is enabled by the null gene approach.[6] When applying parametric spreading, population size is an important consideration. A somewhat larger population size is recommended to accommodate the goal of generating the parametric curve for designated variables and a representation of the Pareto front with the same population.

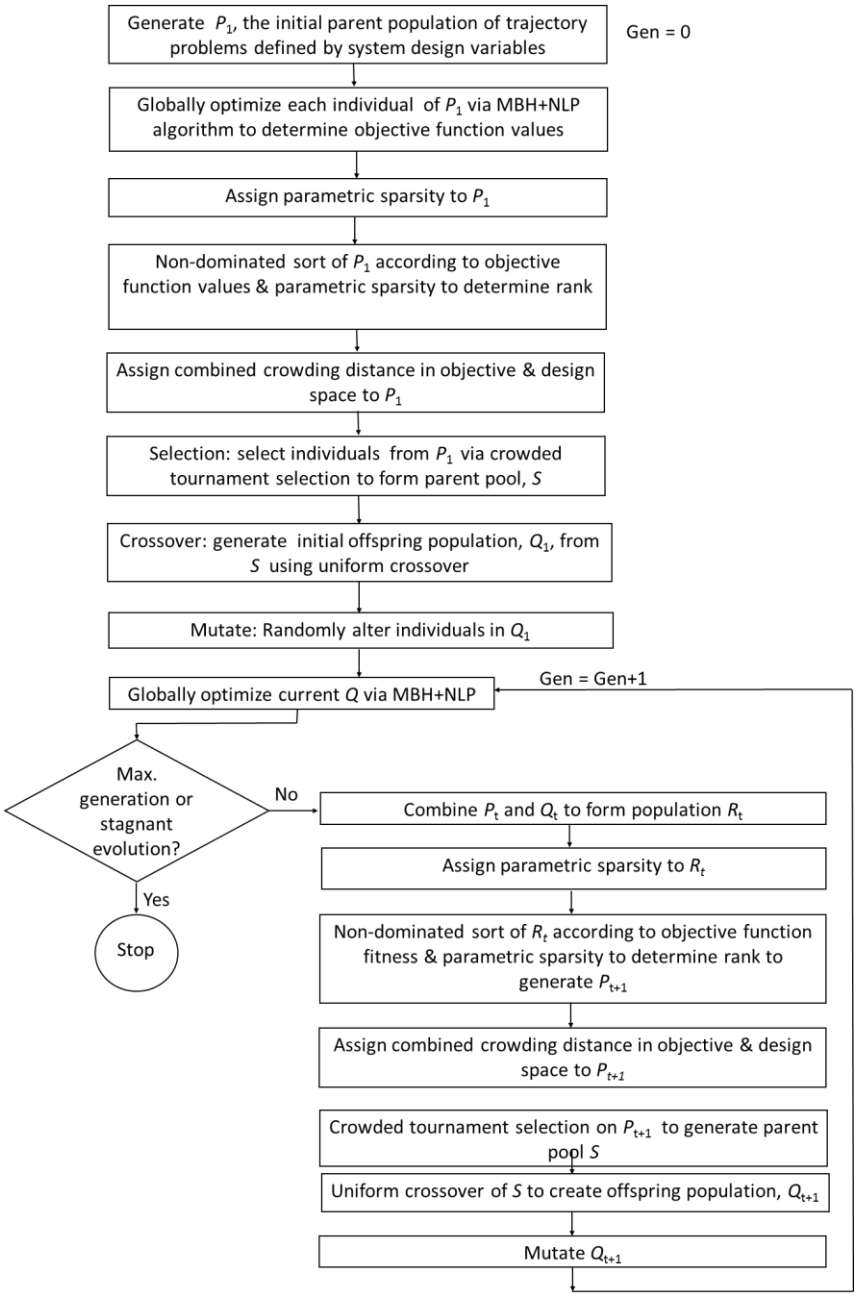

Figure 8: Parametric spreading, multi-objective algorithm based on the NSGA-II

EXAMPLES

The parametric spreading outer loop is applied to two example problems: 1) a high-thrust, variable gravity assist trajectory to an elliptic Neptune orbit, and 2) a systems optimization problem for a low-thrust sample return from Deimos.

High-Thrust Neptune Orbiter Mission Design

The first example for a Neptune orbiter illustrates the multi-objective outer loop GA ability to generate a representation of the Pareto front for two objectives (maximize final mass and minimize TOF) while also incorporating parametric spreading for the launch date. This example is representative of an initial design exploration study to determine mission feasibility. A broad exploration of the trade space is sought, and thus a low-resolution parametric evaluation of launch date is defined with each inner-loop run able to search within 100-day launch period. A six-year range of launch date periods are explored, and the outer loop can vary the flyby sequence with up to five gravity assists using any combination of Venus, Earth, Mars, Jupiter, and Saturn, as outlined in Table 1. A Delta IV Heavy is used as the launch vehicle, and one deep space maneuver per phase is optimized within the inner-loop MGA*n*DSM*s* transcription. Neptune orbit insertion is assumed to occur at a 3000-km altitude into an orbit with an eccentricity of 0.988, with the propellant for the insertion burn accounted in the optimized final mass; other trajectory details are listed in Table . In the outer loop, an NSGA-II population of 256 is used with a 15% mutation rate to encourage population diversity as listed in Table 3. Both a parametric spreading case and standard (i.e., no parametric spreading) case are evaluated for comparison.

Table 1. Outer-loop Outer-Loop Design Variable Menu for Neptune Example

Design Variable	Value	Resolution
Launch period open epoch	[1/1/2024, 1/9/2030}	1 year
Flyby body	{Venus, Earth, Mars, Jupiter, Saturn, null, null, null, null, null, null}	n/a
Flight time	[3200, 7000] days	200 days

Table 2. Trajectory assumptions for Neptune example

Description	Value
Launch period	100 days
Launch declination	[-28.5, 28.5] deg
Launch vehicle curve	Delta IV-H
Chemical I_{sp}	320 s
Neptune arrival date	Determined by optimizer
Neptune insertion orbit semi-major axis	2,346,770 km
Nepturn insertion orbit eccentricity	0.98817
Maximum number of DSMs per phase	1
Inner-loop objective function	Max: $\log_{10}$(final mass)
Inner-loop run time	30 minutes

Table 1. Outer-loop optimization parameters for Neptune Example

Parameter	Value
Population size	256
Mutation probability	15%
Objective Functions	$\log_{10}$(final mass), TOF
Parametric spreading variables	Launch period open epoch

The outer loop for both the parametric spreading case and the regular case is stopped after 100 generations. The outer loop execution cases are numerically independent and can be executed in parallel taking 2.5 days on a 64 node compute cluster comprised of 2.6 GHz AMD Opteron cores. The optimization is fully automated after initial problem definition with no human oversight, and no user-supplied initial guesses. A representation of the two-dimensional Pareto front is shown in Figure 9. The non-dominated members of the final population for the parametric spreading case are represented by diamonds whereas the standard outer-loop case results are represented by an ‘X’. Labeled on the delivered mass versus time of flight projection plot are mission sequence tags representing a sampling of the variety of sequences in the non-dominated set.

Mission sequences along the best non-dominated front include a mixture of flyby sequences: EN (direct), ESN, EEJN, EVEJN, EEMJN. The best performing trajectory in terms of maximum arrival mass is an EEMJN with a 19.1 year TOF, delivering 5000 kg into the eccentric Neptune orbit. The trajectory is plotted in Figure 10.

Notably, the non-dominated fronts from the two cases are very similar. The parametric spreading non-dominated front has a slightly more uniform representation of the Pareto front with optimal solutions at ~10.5 and 12-year flight times, and the standard outer-loop algorithm identifying a slightly better performing solution with a 9.9 year TOF. The color of the solutions in Figure 9 correspond to launch date, and it is apparent that a narrow band of late launch dates are optimal for the longer TOFs. Specifically, a mid-2029 opportunity for the EEMJN solutions and a late-2029 for EEJN and EVEJN solutions dominate the best front.

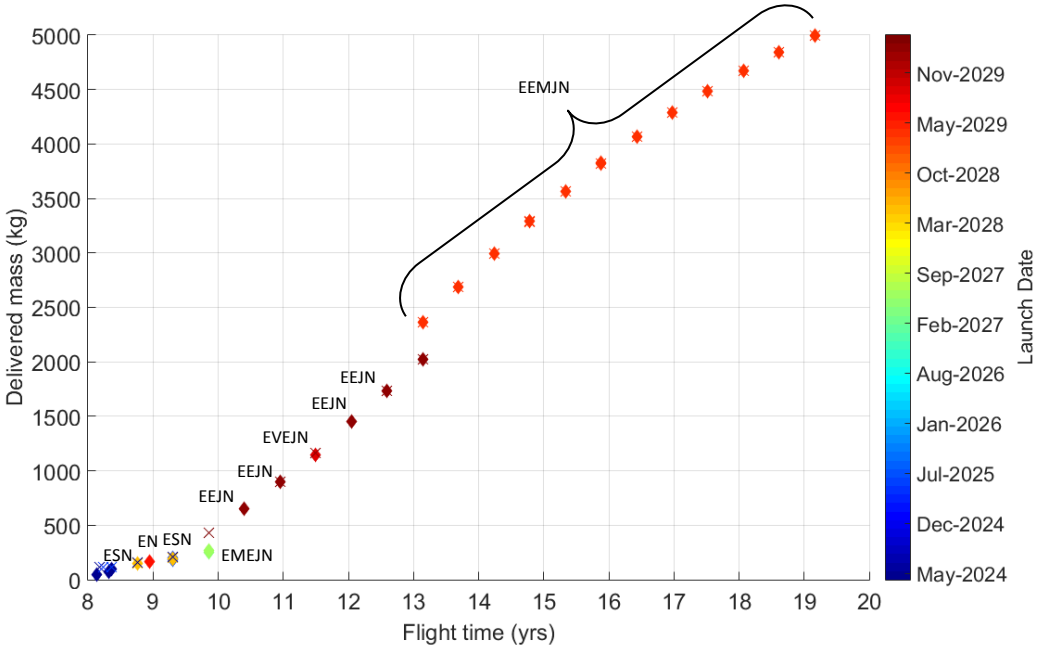

Figure 9: Non-dominated front members in final generation for Neptune orbiter example

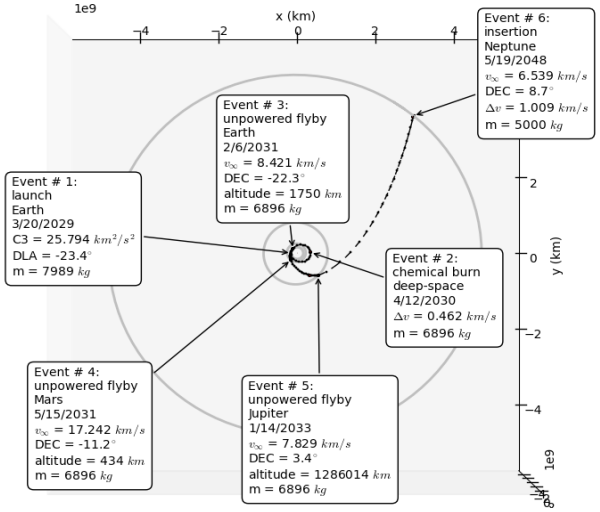

Figure 10: Neptune trajectory with highest final mass (EEMJN)

To evaluate performance of the parametric spreading functionality, the final mass is plotted against launch date for the entire population of both the parametric spreading case and standard case in Figure 11. A line is plotted from the best final mass return for each discrete 100-day launch date period. This line does not represent performance for individual dates, but rather the performance over the discrete outer-loop launch date period variable. Note that solutions often tend to reside at the lower or upper bounds of the launch period. In comparing the blue (parametric spreading) and orange (standard) lines, it is clear that the parametric spreading functionality enables identification of higher delivered mass for each launch date outside of the Pareto-optimal launch period in 2029 and two equivalent performing return masses in 2024 and 2026. For the launch period opening on 6/23/2029 the identified mass by the parametric spreading algorithm is nearly double that identified by the standard outer loop.

The parametric spreading algorithm enables a mission designer better insight in the optimal sensitivity of the final mass to launch date. This capacity is enabled by more diverse exploration in the parametric spreading variable as indicated by the broader spread in launch date of blue diamonds versus orange 'X's in Figure 11. A histogram of the number individuals evaluated in the inner loop through all generations for each discrete launch period is shown in Figure 12. The standard algorithm focuses its search on more runs on more optimal launch period, while the parametric spreading algorithm provides a more uniform exploration for the parametric spreading variable, while also balancing the goal of generating a Pareto front representation with sufficient exploitation of the identified optimal launch periods.

The flight time versus launch date for the entire population with final masses greater than 200 kg is plotted in Figure 13. For this objective both the parametric spreading algorithm and the standard outer loop produce a similar representation of the launch date parametric variation for many of the dates. However, in seven of the possible 23 launch periods, the parametric spreading algorithm identifies a shorter TOF (up to one-year difference), while the standard algorithm identifies a better flight time for one of the 23 periods (i.e., launch period opening on 9/27/2026).

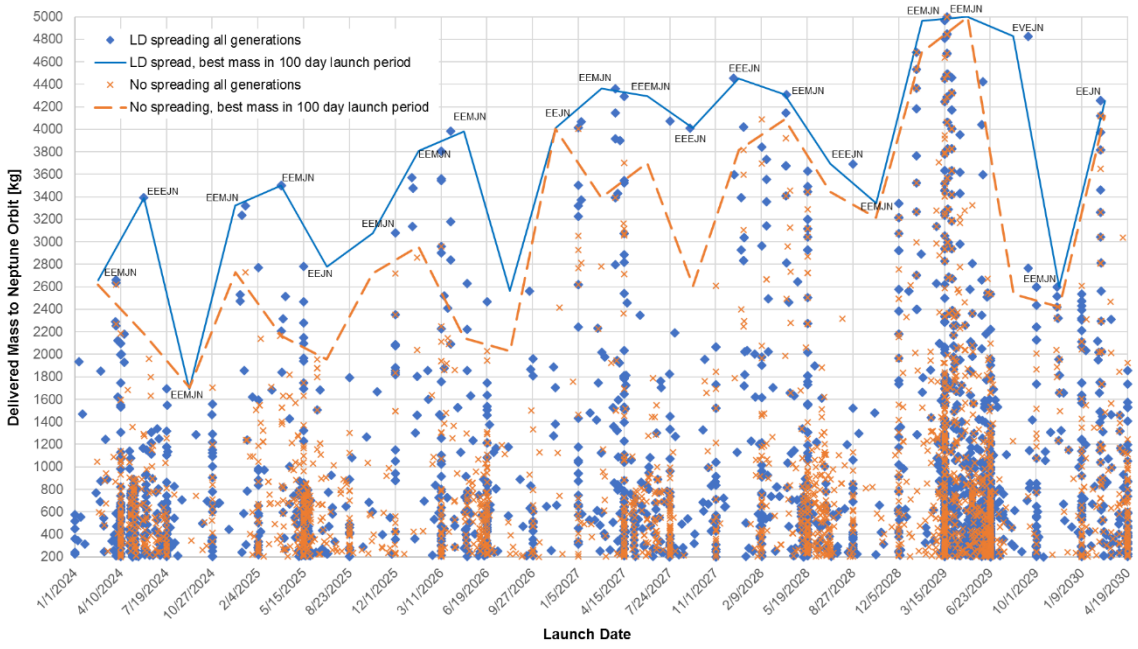

Figure 11: Final mass versus launch date for entire population (blue: parametric spreading, orange: standard algorithm)

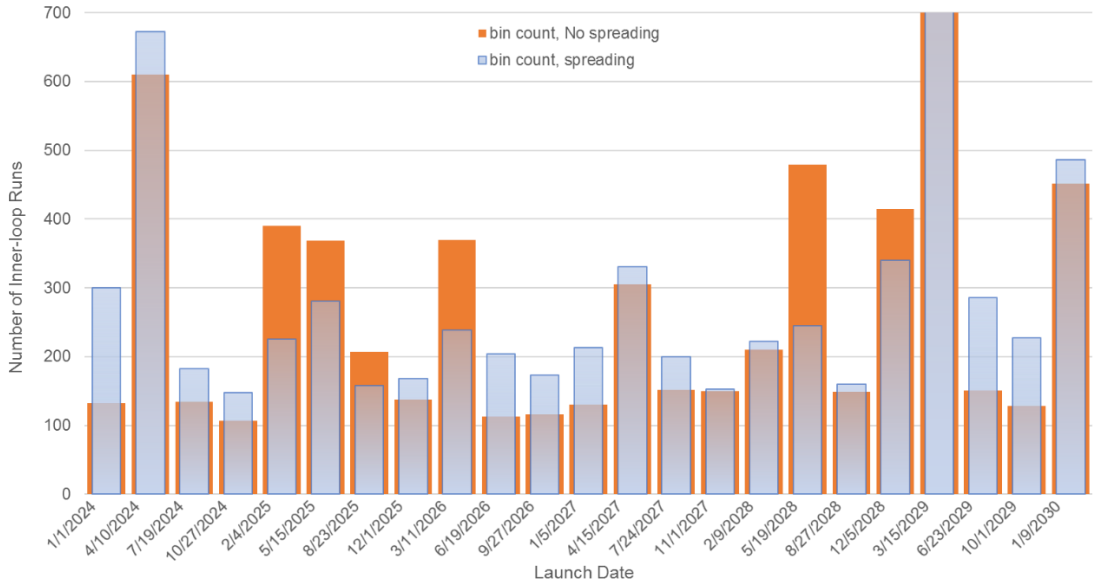

Figure 12: Histogram of number inner-loop evaluations at each launch period for all generation

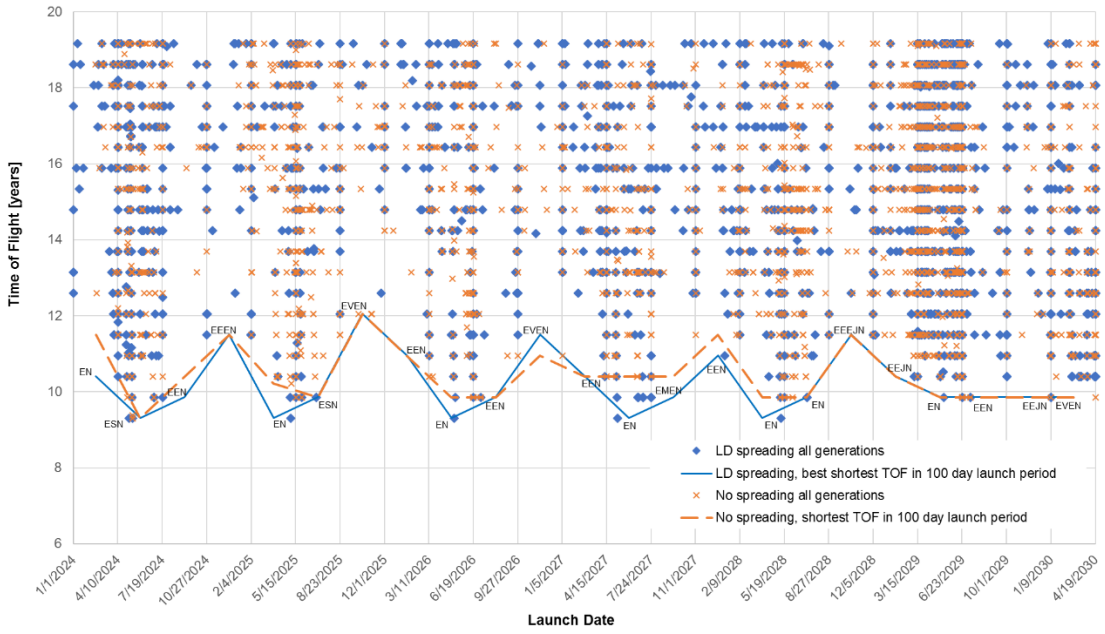

Figure 13: Time of flight versus launch date for entire population (blue: parametric spreading, orange: standard algorithm)

Low-Thrust Multi-Objective, Systems Optimization: Deimos Sample Return

The second example illustrates the outer loop multi-objective systems optimizer to optimize four objectives and produce a representation of the parametric sensitivity of the objectives to launch date variation. The mission aims to return 100 kg of sample mass from Deimos using from one to three NEXT low-thrust engines, with the outer loop varying the Beginning-Of-Life (BOL) power at 1 AU in 2 kW increments from 6 kW to

22 kW. Two discrete versions of the NEXT engine are included as outer-loop variables: a high-thrust version and a high-I_{sp} version. In this example, the dry mass of the spacecraft is not adjusted for the different hardware configurations as it is assumed that the study is occurring before the base spacecraft hardware is known. The outer loop can, however, adjust dry mass according to the hardware selection from the outer-loop menu. The flyby sequence on both outbound and return legs is variable and the outer loop can select up to two flybys using Venus, Earth, or Mars. A low-thrust rendezvous with Earth is executed on the return leg.

The outer loop menu choices are listed in Table 4. The inner-loop and outer-loop optimization parameters are outlined in Table 5, and Table 6, respectively. The low-thrust capture spiral at Mars is modeled using Edelbaum's constant-thrust equation[21] from the radius of Mars' sphere of influence to the orbital radius of Deimos for rendezvous, and vice versa for departure. The algorithm is run for 2.5 days on a 64-core cluster

Table 2: Deimos Sample Return Outer-Loop Menu

Design Variable	Int.	Value	Resolution
Solar array size	[0, 8]	[6, 22] kW	2 kW
Launch period open	[0, 49]	[1/1/2024, 1/29/2028]	30.4 days
Flight time	[0, 26]	[1100, 3000] days	100 days
Engine type	[0, 1]	{NEXT high-I_{sp}, NEXT high-thrust}	-
Number of engines	[0, 2]	[1, 3]	1
Outbound Flyby sequence	[0,6] | [0,6]	{Venus, Earth, Mars, null, null, null}	1
Return Flyby sequence	[0,6] | [0,6]	{Venus, Earth, Mars, null, null, null}	1

to reach 150 generations.

The parametric sensitivity of maximum dry mass versus launch date is illustrated in Figure 14.. A tighter resolution than the Neptune example is available for the launch date parametric spreading variable provided the 30.4 day (~one month) launch period. Additionally, low-thrust trajectories often allow for greater launch date flexibility, but phasing for gravity assists is still critical in launch date performance. All non-dominated solutions are shown in a 2D projection of the objective space in Figure 15 with the BOL power at 1 AU indicated by the color bar. Recall that this representation of the Pareto front is four dimensional as minimization of BOL power, and the number of thrusters are also objectives. These additional two objectives drive the equal-power bands below the best performing solutions in mass and time. As with the Neptune example

Table 5: Inner-Loop Parameters for Deimos Example

Description	Value
Launch period	30.4 days
Wait time at Deimos	[250, 350] days
Min. spacecraft mass	500 kg
Maximum launch C_3	Atlas V 551 limit
Departure declination (EME2000)	[-28.5, 28.5] deg
Thruster duty cycle	90%
Max. number of SNOPT major iterations	8000
Power available	$1/r^2$
Spacecraft bus power	2 kW
Propellant margin	none
Inner-loop run time	12 minutes
Inner-loop objective	Max. final mass

Table 6: Outer-Loop Optimization Parameters

Parameter	**Value**
Population size	180
Mutation probability	15%
Objective Functions	$\log_{10}$(final mass), TOF
Parametric spreading variables	Launch period open poch (30.4 day period)

both a representation of the Pareto front is achieved while also providing the parametric sensitivity of the objectives to launch date.

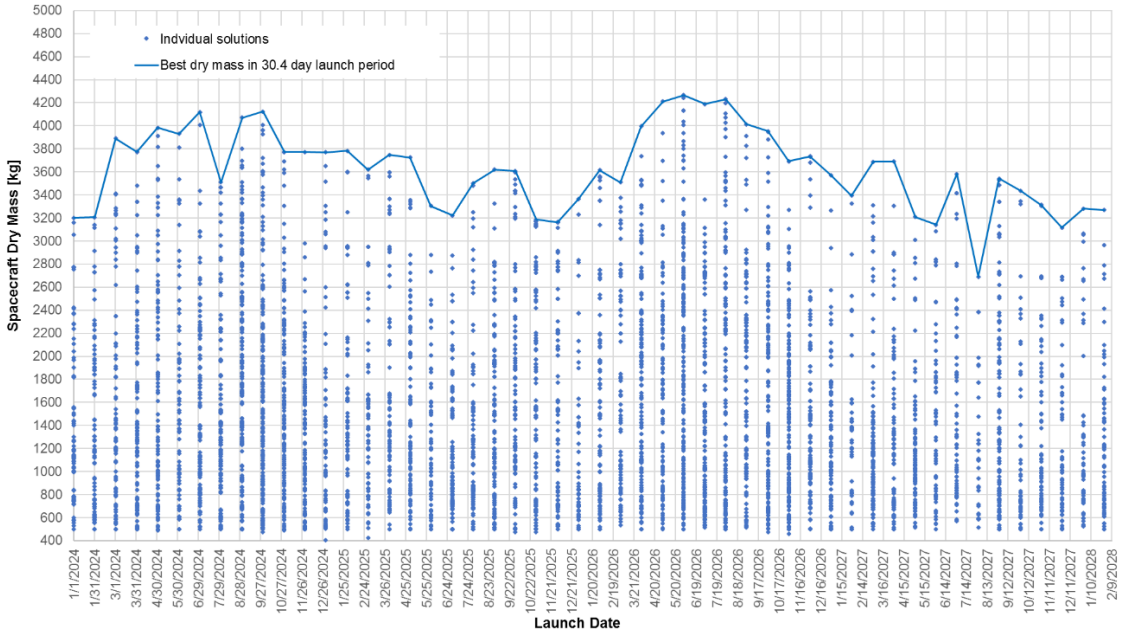

Figure 14: Dry mass versus launch date for all generations, Deimos example

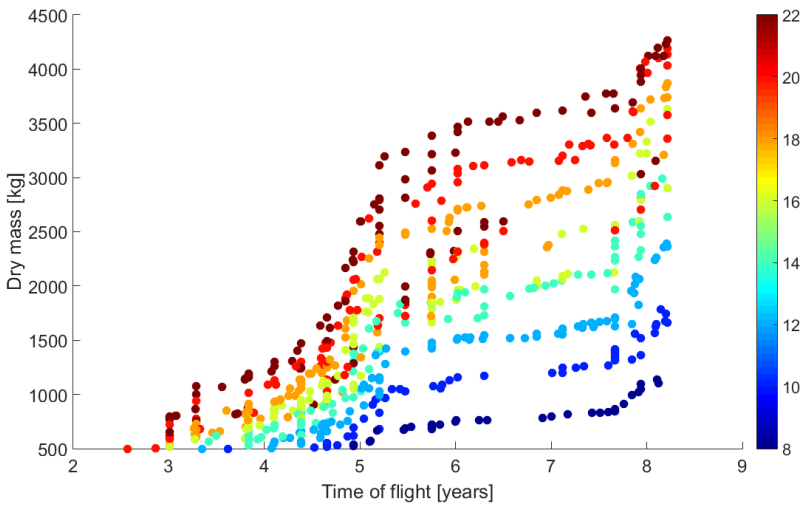

Figure 15: Best non-dominated front for Deimos example colorized by BOL power (kw) at 1AU kw

MULTI-OBJECTIVE VISUALIZATION

A key challenge in multi-objective trajectory and systems optimization is visualizing the rich, multi-dimensional solution space and its relationship to an expansive and complex design space. To enable a better understanding of the solution set, visualization tooling is developed to enable illustration of solution and design space relationships and ease communication of the results to other team members. In the visualization tooling, multi-dimensional, Pareto-optimal solutions and the dominated solutions retained for parametric

studies are plotted in a variety of plot combinations, dynamically mapping the design and solution space relationships. Additionally, the thousands of solutions evolved throughout the generations of the genetic algorithm can provide further insight into the problem. The visualization interface allows for viewing of both 2D and 3D color coded data sets, while providing easy zooming and user-specified filtering. The user is able to interact with any data point in the trade space, linking detailed information about the particular solution, both hardware and trajectory information, as well as trajectory plots.

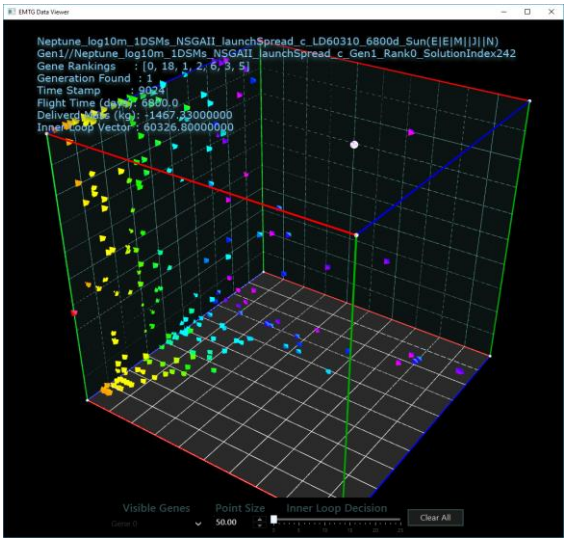

Figure 16: Multi-objective optimization data visualization, dynamically plotted and colorized gene parameters in 3D

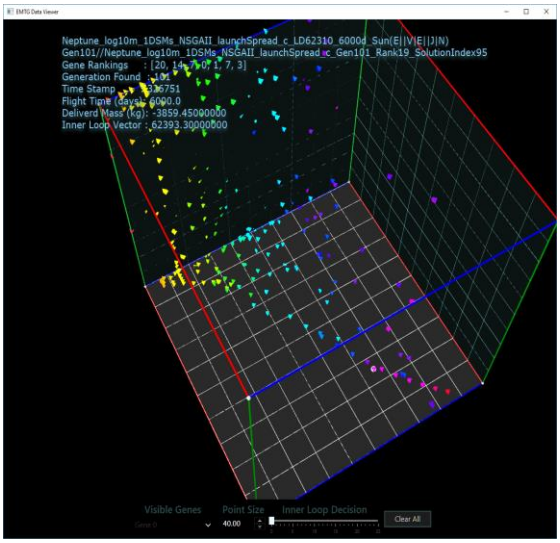

Figure 17: Final generation 100 stacked into same optimization data as Figure 16 to show

Figure 16 displays the Neptune orbiter example optimization data when filtered for all items where Gene0 (Launch window open date) is ranked 0 (Best). The X axis is a numerically scaled Delivered Mass to Orbit (kg). The Y axis is flight time (days). The Z axis is the primary inner loop decision variable, in this scenario the Launch Window open date. The color scheme selected in these views is a color wheel style where red maps to the lowest value of the launch window open date and dark purple maps to the highest such value. This effectively connects the color range to the Z axis which is useful visually but can be changed using the tool. Figure 17 shows the same primary data set as Figure 16 but with the final generation 100 data stacked on top. This stacking is facilitated through a drag and drop interface and allows the user to see multiple evolutionary generations in concert.

CONCLUSION

To enable simultaneous optimization of multiple objectives and generation of objective function sensitivity to key design variables, a multi-objective genetic algorithm that incorporates parametric spreading is presented. The parametric spreading algorithm is incorporated in the outer loop of EMTG, and operates in concert with the MBH+NLP inner-loop to solve hybrid optimal control problems. The user can specify any number of objectives and any number outer-loop design variables for parametric spreading. The parametric spreading algorithm generates a representation of the Pareto front and enables evaluation of parametric sensitivity to the specified design variables. The parametric spreading algorithm is based on the NSGA-II, and incorporates additional diversity preservation mechanisms to balance clustering in design space while also maintaining strong exploratory capability in the parametric spreading variable. The algorithm has broad applicability to high- or low-thrust interplanetary design problems. A variety of mission types can be explored including multi-flyby scenarios, and optimization of spacecraft systems variables can also be incorporated.

REFERENCES

[1] Pareto, V. *Manuale di economica politica, societa editrice libraria*, Milano, Italy, 1906; translated into English A. Schwier as *Manual of Political Economy*, MacMillan Press, New York, 1971

[2] Hartmann, J.W., Coverstone-Carroll, V., and Williams, S. N., "Optimal Interplanetary Spacecraft Trajectories via a Pareto Genetic Algorithm," Journal of the Astronautical Sciences, Vol. 46, No. 3, 1998, pp. 267–282

[3] Vavrina, M.A., Howell, K.C., "Global Low Thrust Trajectory Optimization through Hybridization of a Genetic Algorithm and a Direct Method," AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii, August 18-21 2008.

[4] Vavrina, M. A., Englander, J. A., and Ghosh, A., "Coupled Low-Thrust Trajectory and Systems Optimization via Multi-objective Hybrid Optimal Control," AAS/AIAA Space Flight Mechanics Meeting Williamsburg, VA, January 2015.

[5] Englander, J. A., Vavrina, M. A., Ghosh, A.R., "Multi-objective Hybrid Optimal Control for Multiple-Flyby Low-Thrust Mission Design," AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA, January 2015.

[6] Englander J.A., and Conway B., "An Automated Solution of the Low-Thrust Interplanetary Trajectory Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 1 (2017), pp. 15-27.

[7] Gill, P.E., Murray, W., and Saunders, M.A., "SNOPT: An SQP algorithm for large-scale constrained optimization." SIAM journal on optimization, Vol. 12 No. 4, 2002, pp. 979-1006.

[8] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, April 2002, pp. 182-197, 2002.

[9] Vavrina, M.A., Englander J.A., Ellison, D.H., "Global Optimization of n-Maneuver, High-Thrust Trajectories Using Direct Multiple Shooting," 26th AAS/AIAA Space Flight Mechanics Conference, Napa, CA, 2016.

[10] Uphoff, C., "Orbit Design Concepts for Jupiter Orbiter Missions," AIAA Mechanics and Control Conference, AIAA Paper 74-781, Anaheim, CA, August 1974.

[11] Sims, J.A., and Flanagan, S.N., "Preliminary Design of Low-Thrust Interplanetary Missions," Paper AAS 99-338, AAS/AIAA Astrodynamics Specialist Conference, Girdwood, AK, August 16-18, 1999.

[12] Sims, J.A., Finlayson, P. A., Rinderle, E.A., Vavrina, M.A., and Kowalkowski, T.D., "Implementation of a Low-Thrust Trajectory Optimization Algorithm for Preliminary Design," Paper AIAA-2006-6746, AIAA/AAS Astrodynamics Specialist Conference, Keystone, CO, August 21-24, 2006.

[13] McConaghy, T.T., "Design and Optimization of Interplanetary Spacecraft Trajectories," Ph.D. Dissertation, School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, 2004.

[14] "PaGMO (Parallel Global Multiobjective Optimizer)," http://pagmo.sourceforge.net/pagmo/index.html.

[15] Vavrina, M.A., and Howell, K.C., "Global Low Thrust Trajectory Optimization through Hybridization of a Genetic Algorithm and a Direct Method," AIAA/AAS Astrodynamics Specialist Conference, Honolulu, HI, August 2008.

[16] Marler, R. Timothy, and Jasbir S. Arora. "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, Vol. 26 No. 6, 2004, pp. 369-395.

[17] Zadeh, L., "Optimality and non-scalar-valued performance criteria," IEEE Transactions on Automatic Control, Vol. 8, No. 1, 1963, pp. 59–60.

[18] Holland, J.H., "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Computing*, Vol. 2 No. 2 1973, pp. 88-105.

[19] Coello, C. A., Lamont, G. B., and Veldhuizen, D. A., *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, 2nd ed., Springer-Verlag, New York, 2006.

[20] Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, 2001.

[21] Edelbaum, T., "Propulsion Requirements for Controllable Satellites," *ARS Journal*, Vol. 31, No. 8, 1961, pp. 1079-1089.